This is AutoGen, an automated text file generator. It was inspired out of frustration and hassle with maintaining syncronization between option flag lists, global variables and usage information. The desire for more than #define macros came about when it became apparent that macros alone were insufficient for reducing the maintenance into a single option list. The impetus to actually start something finally came when I had to maintain a large callout procedure table and associated lookup tables.

Rev 1 of this utility was a set of #define macro expansions.
Rev 2 was a shell script that sort-of did a prototype.
    Much better than just #defines, but still clearly lacking.
Rev 3 had a very kludgy macro definition syntax.
Rev 4 a reworking and simplification of the declarations
Rev 5 the addition of Guile expression processing

Mailing lists can be found on SourceForge:

autogen-users@lists.sourceforge.net

*** AutoGen requires: ***

1. POSIX regular expression library. If not available by default, use the --with-regex-* options to specify how to find, use and link to it.
2. an ANSI C compiler
3. The Guile version of a Scheme processing language.

It

*** Installation note: ***

AutoGen does *NOT* contain any compiled-in configuration information. Therefore, in order to use the templates that come bundled with it, you must tell AutoGen how to find those templates when you build applications that use those templates.

1. by doing nothing. If you do not alter the default data directory, AutoGen will search for templates in the directory ../share/autogen, relative to the executable directory. That should generally work.

2. You can tell AutoGen where to look with an environment variable:

```
export AUTOGEN_TEMPL_DIR=$prefix/share/autogen
```

3. You can use an RC file:

```
autogen -L $prefix/share/autogen --save=$HOME/.autogenrc
```

4. If you have an old Guile library, you will find that its error reporting does not work so well. Consequently, you will see "make check" failures in the output text where you would expect to find file name and line number references for the invalid input. Please upgrade your Guile lib.

5. You can build and install AutoGen, ensuring you have the automake/autoconf/libtool-s needed, and then editing agen5/opts.def thus:

```
echo "homerc = $prefix/share/autogen;" >> agen5/opts.def
```

and then rebuilding AutoGen. However, if you do this latter "fix", you will have an immobile product. I hate that, others like it. It is, however, up to you.

*** Build note: ***

Sometimes, configure believes it has done a good job when it really hasn't. It is possible to configure a system in such a way that the Guile headers and libguile are linked against correctly, but the loader cannot find libguile.so.xxx. This is because GCC will silently find the library for linking, but not set the library dependencies correctly. The consequence is that the configure script believes that standard links will produce working executables. It won't. The simplest solution is:

```
.../configure --disable-shared
```

the best solution is to examine the output of ``guile-config link'', duplicate the ``-L/path/to/lib'' argument, but changing the "-L" to "-R" or "-Wl,-rpath," or "-rpath" or whatever it happens to be that works for your platform, and hand that off to configure as the argument to ``--with-libguile-link''. "libtool" won't fix it and it's too hard for me. Sorry. Anyway, for example, assume:

```
guile-config link
```

produces:

```
  -L/opt/sfw/lib -lguile -lm
```

now run configure as follows:

```
  .../configure \
    --with-libguile-link='-L/opt/sfw/lib -lguile -lm -R/opt/sfw/lib'
```

Isn't that special?

*** Bootstrap note: ***

I have some private tools referenced in the various bootstrap scripts.
Unless you have these tools, bootstrap won't work for you. I intend
to fix this as time permits. Meanwhile, there is also a tarball in CVS
of all the bootstrap-generated files:
  noag-boot.sh

*** Licensing: ***

autogen is under GPL, but that does not cause the output produced by autogen to be under GPL. The reason is that the output contains next to nothing that comes directly from autogen's source code. Thus, the output is not a "derivative work" of autogen (in the sense of U.S.@: copyright law). The output is primarily a derivative of the input templates.

On the other hand, the output produced by autogen contains essentially all of the input template. Therefore the output is under the same license, with the same copyright holder, as the input that was passed to autogen.